

# 基于组合加速机制的多特定类快速正域约简

张小飞

烟台大学计算机与控制工程学院, 山东 烟台

收稿日期: 2023年5月27日; 录用日期: 2023年6月27日; 发布日期: 2023年7月5日

## 摘要

信息技术的快速发展给社会带来了海量的高维数据, 这些海量数据中隐藏着大量有价值的信息。如何高效处理大规模数据并从中提取有效知识已成为计算机领域的研究热点。基于粗糙集理论的属性约简, 可以在保证数据分类能力不变的前提下, 删除冗余属性, 从而实现数据的有效降维。在实际应用中, 决策者往往只关注某些特定决策标签的有效信息提取。在多特定类属性约简中, 传统的启发式算法约简效率较低。针对该问题, 本文从对象、属性和粒度的视角出发, 提出了基于组合加速机制的多特定类快速正域约简算法。最后, 实验选取6组数据集进行实验, 从约简长度、参与迭代的对象规模、迭代次数和约简时间四个方面验证了所提算法在多特定类约简中的有效性。

## 关键词

粗糙集, 属性约简, 粒计算, 加速机制

## A Fast Attribute Reduction Algorithm Based on Fusing Acceleration Mechanism for Multi-Specific Classes Positive Region

Xiaofei Zhang

School of Computer and Control Engineering, Yantai University, Yantai Shandong

Received: May 27<sup>th</sup>, 2023; accepted: Jun. 27<sup>th</sup>, 2023; published: Jul. 5<sup>th</sup>, 2023

## Abstract

The rapid development of information technology has brought massive high-dimensional data to society, which hides a large amount of valuable information. How to efficiently deal with these large-scale data and extract effective knowledge from it has become a research hotspot in the field

of computer science. Attribute reduction based on rough set theory can remove redundant attributes while keeping the ability of data classification unchanging, thus reducing the dimension of data effectively. In practical applications, decision makers often only focus on the effective information extraction of certain specific decision labels. In the attribute reduction of multi-specific classes, traditional heuristic algorithms have lower reduction efficiency. To solve above problems, this paper proposes a fast attribute reduction algorithm based on fusing acceleration mechanism for multi-specific classes positive region, which is from the perspectives of objects, attributes and granularity. Finally, six datasets were used for experiments. And the experimental results show the effectiveness of the proposed accelerating algorithm in this paper for multi-specific decision classes attribute reduction, which is verified from four aspects: reduction length, size of objects in iterations, number of iterations and reduction time.

## Keywords

Rough Set, Attribute Reduction, Granular Computing, Acceleration Mechanism

Copyright © 2023 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

粗糙集理论[1][2]是一种处理不精确、不一致信息的有效数学工具,具有极为广泛的应用场景,目前已经成功应用于智能工业[3][4]、图像处理[5][6]、智能决策[7]等多个领域。

属性约简[8][9]是粗糙集理论研究中的关键课题之一。属性约简的核心思想是在保持分类能力不变的前提下,删除冗余属性,从而实现海量数据的有效降维。基于贪心策略的启发式算法是获取属性约简的主要方法之一。传统的启发式属性约简算法效率较低,难以有效地处理大规模数据,为了进一步提高算法的约简效率,许多学者从多种角度对启发式算法的加速机制展开了研究。Qian 等[10]提出了正向近似的加速策略,通过在算法迭代过程中逐步删除正域的方式,有效提高了算法效率。陈曼如等[11]从样本和属性的视角出发,通过减少正域样本和算法的迭代次数,设计了基于正区域不变的快速属性约简算法。赵立威等[12]在序决策系统下通过优化启发式算法中的属性添加策略,构建了基于特征粒的快速约简算法。Chen 等[13]从对象、属性和粒度视角设计了基于决策系统全部分类的加速机制,提出了基于融合加速机制的通用属性约简加速算法框架。

以上算法都是针对决策系统中全部决策类进行属性约简,但在实际应用中,决策者往往对某一个或多个特定类决策类更感兴趣。为此许多学者针对特定类的属性约简问题展开了研究并取得了很多优秀的研究成果。Liu 等[14]在多种约简目标下建立了基于差别矩阵方法的特定类属性约简,为局部属性约简提供了新的思路。Zhang 等[15]从信息论和代数角度分析了全部决策类与特定类约简之间的层次联系。Wang 等[16]在邻域决策系统中,针对特定类提出了局部条件熵的定义,并设计了基于条件熵的局部属性约简算法。

虽然有大量学者针对启发式算法进行了优化,但在多特定类的属性约简研究中,现有的经典启发式属性约简算法仍存在效率较低的问题,难以有效处理大规模数据。针对该问题,本文从多特定类的视角出发,提出了多特定类正域约简的定义,并结合对象、属性和粒度的加速方法,设计了基于组合加速机制的多特定类快速正域约简算法,并通过实验验证了所提算法的有效性。

## 2. 基本概念

给定一个 Pawlak 决策系统  $S=(U,C\cup D)$ ，其中， $U$  为对象集合，称为论域， $C$  和  $D$  分别为条件属性集和决策属性集。 $f(u_i, a_t)$  表示对象  $u_i$  在属性  $a_t$  上的取值。例如表 1 所示的 Pawlak 决策系统中，论域  $U = \{u_1, u_2, u_3, u_4\}$ ，条件属性集  $C = \{a_1, a_2, a_3\}$ ，决策属性  $D = \{d\}$ 。

**Table 1.** Pawlak decision system  
**表 1.** Pawlak 决策系统

$U$	$a_1$	$a_2$	$a_3$	$d$
$u_1$	2	1	2	1
$u_2$	2	0	1	2
$u_3$	1	2	0	2
$u_4$	0	1	1	3

**定义 1 [1]** 给定一个决策系统  $S=(U,C\cup D)$ ，对于  $\forall A \subseteq C$ ，定义  $A$  上的不可区分关系为：

$$R_A = \{(u_i, u_j) \in U^2 \mid \forall a_t \in A, f(u_i, a_t) = f(u_j, a_t)\}.$$

$R_A$  是一个满足自反、对称和传递的等价关系，根据等价关系  $R_A$  可导出论域上的一个划分  $U/R_A = \{[u_i]_A \mid u_i \in U\}$ ，其中， $[u_i]_A = \{u_j \in U \mid (u_i, u_j) \in R_A\}$  表示对象  $u_i$  关于属性集  $A$  的等价类，为方便表示， $U/R_A$  可简记为  $U/A$ 。基于决策属性  $D$  对论域形成的划分为  $U/D = \{D_1, D_2, \dots, D_m\}$ ，称为决策类集合，集合  $D_{mcs} \subseteq U/D$  ( $1 \leq |D_{mcs}| \leq |U/D|$ ) 称为多特定类。下面给出多特定类  $D_{mcs}$  上下近似的定义。

**定义 2 [1]** 给定一个决策系统  $S=(U,C\cup D)$ ， $D_{mcs} \subseteq U/D$ ，对于  $\forall A \subseteq C$ ，多特定类  $D_{mcs}$  关于属性集  $A$  的上下近似集合分别为：

$$\begin{aligned} \overline{R}_A(D_{mcs}) &= \bigcup_{D_k \in D_{mcs}} \{[u_i]_A \mid [u_i]_A \cap D_k \neq \emptyset\}, \\ \underline{R}_A(D_{mcs}) &= \bigcup_{D_k \in D_{mcs}} \{[u_i]_A \mid [u_i]_A \subseteq D_k\}. \end{aligned}$$

基于上下近似的概念，以多特定类  $D_{mcs}$  为目标概念，通过粗糙集理论可以将论域划分为正域、边界域和负域，定义分别为：

$$\begin{aligned} POS_A(D_{mcs}) &= \underline{R}_A(D_{mcs}), \\ BND_A(D_{mcs}) &= \overline{R}_A(D_{mcs}) - \underline{R}_A(D_{mcs}), \\ NEG_A(D_{mcs}) &= U - \overline{R}_A(D_{mcs}). \end{aligned}$$

由于  $D_{mcs} \subseteq U/D$  且  $1 \leq |D_{mcs}| \leq |U/D|$ ，因此多特定类具有良好的扩展性，多特定类既可以包含一个决策类也可以包含多个甚至全部决策类。

## 3. 多特定类正域约简

在实际应用中，决策者往往更偏好于某一个或多个特定决策标签的有效知识提取，本节将介绍多特定类正域约简的相关概念和算法。

**定义 3** 给定一个决策系统  $S=(U,C\cup D)$ ， $D_{mcs} \subseteq U/D$ ， $\forall A \subseteq C$ ，对于  $\forall a \in A$ ，属性  $a$  的内部属性

重要度为:

$$Sig_{in}(a, A, D_{mcs}) = |POS_A(D_{mcs})| - |POS_{A-\{a\}}(D_{mcs})|.$$

**定义 4** 给定一个决策系统  $S=(U, C \cup D)$ ,  $D_{mcs} \subseteq U/D$ ,  $\forall A \subseteq C$ , 对于  $\forall a \in C - A$ , 属性  $a$  的外部属性重要度为:

$$Sig_{out}(a, A, D_{mcs}) = |POS_{A \cup \{a\}}(D_{mcs})| - |POS_A(D_{mcs})|.$$

内部属性重要度可以用来评估属性是否冗余, 常用于求核和去冗余操作。外部属性重要度往往用于迭代过程中的最佳属性选择。

**定义 5** 给定一个决策系统  $S=(U, C \cup D)$ ,  $D_{mcs} \subseteq U/D$ , 若  $A \subseteq C$  为多特定类  $D_{mcs}$  的正域约简, 则满足如下条件:

- 1)  $POS_A(D_{mcs}) = POS_C(D_{mcs})$ ;
- 2)  $\forall A' \subset A, POS_{A'}(D_{mcs}) \neq POS_C(D_{mcs})$ 。

其中, 条件 1)保证了约简前后多特定类正域不发生变化, 条件 2)则保证了约简属性集中的任一属性都是必要的。基于正向贪婪的多特定类启发式属性约简算法如表 2 所示。该算法的流程图如图 1 所示。

在算法 FGARM 中, Step1 为初始化过程; Step2 为迭代选取属性的过程, 根据外部属性重要度对属性进行评估, 每次选择一个最佳属性添加至约简属性集中, 直到满足中止条件为止; Step3 和 Step4 为去冗余和输出约简的过程。

**Table 2.** A forward greedy heuristic attribute reduction algorithm for multi-specific decision classes (FGARM)

**表 2.** 基于正向贪婪的多特定类启发式属性约简算法

输入: 决策系统  $S=(U, C \cup D)$ 。

输出: 多特定类  $D_{mcs}$  的一个正域约简  $A$ 。

Step1:  $A = \emptyset$ ;

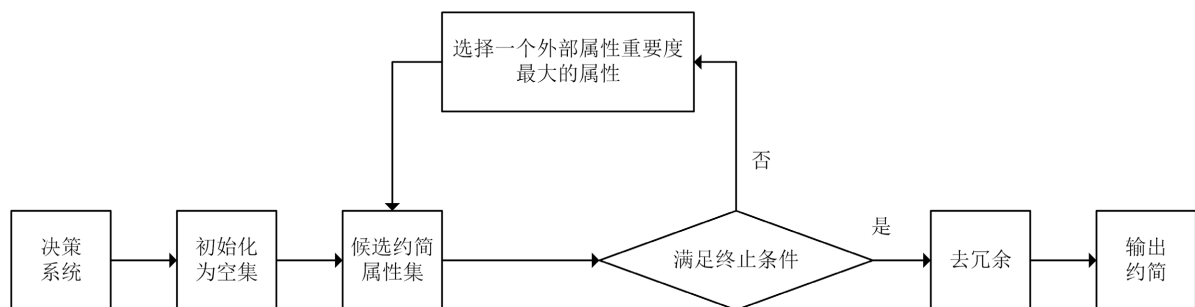
Step2: 若  $|POS_A(D_{mcs})| \neq |POS_C(D_{mcs})|$ ;

Step2.1: 对于  $\forall b \in C - A$ , 计算  $Sig_{out}(b, A, D_{mcs})$ ;

Step2.2: 令  $b_1 = \arg \text{Max}\{Sig_{out}(b, A, D_{mcs})\}$ ,  $A = A \cup \{b_1\}$ ;

Step3: 对于  $\forall a \in A$ , 如果  $Sig_{in}(a, A, D_{mcs}) = 0$ , 则令  $A = A - \{a\}$ ;

Step4: 返回约简  $A$ , 算法结束。



**Figure 1.** A flow chart of FGARM algorithm

**图 1.** 算法 FGARM 流程图

#### 4. 多特定类快速正域约简

算法 FGARM 在计算约简时存在待处理对象规模较大、属性迭代次数过多和待评估属性数目较多等问题，难以高效处理大规模数据。为进一步提高多特定类的约简效率，本节从对象、属性和粒度三个方面分析了多特定类启发式算法的加速机制，提出了基于组合加速机制的多特定类快速正域约简算法。

**定理 1** 给定一个决策系统  $S = (U, C \cup D)$ ,  $D_{mcs} \subseteq U/D$ ,  $\forall A \subseteq C$ , 对于  $\forall a \in C - A$ , 满足:

$$POS_{A \cup \{a\}}^U(D_{mcs}) = POS_A^U(D_{mcs}) \cup POS_{A \cup \{a\}}^{U'}(D_{mcs}),$$

其中,  $U' = U - POS_A^U(D_{mcs}) - NEG_A^U(D_{mcs})$ 。

定理 1 表明, 随着属性的增加, 边界域中的部分样本会被精确划分至正域, 因此可以通过删除正域和负域的方式实现多特定类的对象加速。当所选多特定类为全部决策类时, 此时负域为空集, 此时算法退化到文献[10]中的情况, 每次迭代过程中仅删除正域中的对象。

**定义 6** 给定一个决策系统  $S = (U, C \cup D)$ ,  $D_{mcs} \subseteq U/D$ , 属性集  $A \subseteq C$ ,  $AT = C - A$ , 对于  $\forall b_i \in AT$ ,

若  $\left| POS_{\{b_1\} \cup \{b_2\}}(D_{mcs}) \right| \geq \left| POS_{\{b_1\} \cup \{b_3\}}(D_{mcs}) \right| \geq \dots \geq \left| POS_{\{b_1\} \cup \{b_{|AT|}}(D_{mcs}) \right|$  且满足

$$\bigcup_{i=1}^2 POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) - POS_{\{b_1\}}(D_{mcs}) \neq \emptyset,$$

$$\bigcup_{i=1}^3 POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) - \bigcup_{i=1}^2 POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) \neq \emptyset,$$

...

$$\bigcup_{i=1}^k POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) - \bigcup_{i=1}^{k-1} POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) = \emptyset,$$

$$\bigcup_{i=1}^{k+1} POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) - \bigcup_{i=1}^k POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) \neq \emptyset,$$

...

$$\bigcup_{i=1}^{|AT|} POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) - \bigcup_{i=1}^{|AT|-1} POS_{\{b_1\} \cup \{b_i\}}(D_{mcs}) \neq \emptyset,$$

则称  $Group(b_1) = \{b_1, b_2, \dots, b_{k-1}, b_{k+1}, \dots, b_{|AT|}\}$  为属性  $b_1$  的属性组。

定义 6 为属性组的定义, 在启发式算法的迭代过程中, 可以通过构造属性组的方式向约简属性集中一次添加多个属性, 降低算法的迭代次数, 实现多特定类的属性加速。同时, 通过差集运算, 有效地避免了部分冗余属性的存在。

**定义 7** 给定一个决策系统  $S = (U, C \cup D)$ ,  $D_{mcs} \subseteq U/D$ ,  $\forall A \subseteq C$ , 多特定类  $D_{mcs}$  关于属性子集  $A$  的局部粒度为:

$$G_A = \frac{|\Omega(A, D_{mcs})|}{|U|^2},$$

其中,  $\Omega(A, D_{mcs}) = \{(u_i, u_j) \in U^2 \mid u_i, u_j \notin NEG_A(D_{mcs})\}$ 。

定义 7 为局部粒度的定义, 由于负域中的对象与多特定类中的决策规则无关, 因此可以仅利用正域和边界域中的样本对局部粒度进行定义。属性的局部粒度值越大, 则该属性对多特定类样本的区分能力越弱, 在属性评估过程中, 可以通过忽略一些粒度较粗的属性以降低待评估的属性规模, 实现多特定类的粒度加速。当所选多特定类为全部决策类时, 此时负域为空集, 则局部粒度退化到文献[13]中的全局粒度定义, 此时粒度可用于表示属性对决策系统中所有样本的区分能力。

通过上述定义，可以在启发式算法中设计基于对象、属性和粒度的多特定类组合加速机制，提高算法的约简效率。下面给出基于组合加速机制的多特定类快速正域约简算法，如表 3 所示。该算法的流程图如图 2 所示。

**Table 3.** A fast attribute reduction algorithm for multi-specific decision classes positive region based on fusing acceleration mechanism (FARAM)

**表 3.** 基于组合加速机制的多特定类快速正域约简算法

输入：决策系统  $S=(U, C \cup D)$ 。

输出：多特定类  $D_{mcs}$  的一个正域约简  $A$ 。

Step1:  $A = \emptyset, T = \emptyset, isFirst = 1$ ;

Step2: 对于  $\forall a \in C$ ，计算局部粒度值  $G_{\{a\}}$ ;

Step3: 若  $|POS_A^U(D_{mcs})| \neq |POS_C^U(D_{mcs})|$ :

Step3.1: 若  $isFirst = 1$ ，则令  $T = C, isFirst = 0$ ，转到 Step3.4;

Step3.2: 若  $isFirst = 0$ ，则令  $T = \{a | \forall a \in C - A, G_{\{a\}} \leq G_{\{b_i\}}\}$ ;

Step3.3: 若  $T = \emptyset$ ，则令  $T = C - A$ ;

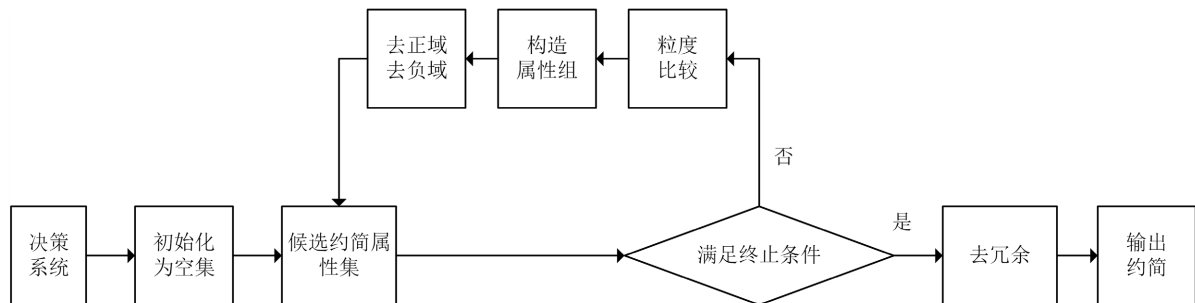
Step3.4: 对  $\forall b \in T$ ，计算  $Sig_{out}(b, A, D_{mcs})$ ，令  $b_i = \arg \text{Max}\{Sig_{out}(b, A, D_{mcs})\}$ ;

Step3.5: 对于  $\forall b_i \in C - A$ ，构造属性组  $Group(b_i)$ ;

Step3.6: 令  $A = A \cup Group(b_i), U = U - POS_A(D_{mcs}) - NEG_A(D_{mcs})$ ;

Step4: 对于  $\forall a \in A$ ，如果  $Sig_{in}(a, A, D_{mcs}) = 0$ ，则令  $A = A - \{a\}$ ;

Step5: 返回约简  $A$ ，算法结束。



**Figure 2.** A flow chart of FARAM algorithm

**图 2.** 算法 FARAM 流程图

在算法 FARAM 中，Step1 为初始化过程；Step2 为计算各属性的局部粒度值，用于后续迭代过程中的粒度比较；Step3 为迭代选取属性的过程，其中，Step3.1~Step3.3 为粒度比较以降低待评估属性数目，Step3.4~Step3.5 为构造属性组实现属性加速的过程，Step3.6 为删除正域和负域实现对象加速的过程；Step4 和 Step5 为去冗余和输出约简的过程。

相比于算法 FGARM 中，算法 FARAM 的优势为：

1) 算法 FARAM 在每轮迭代过程中通过删除正域和负域的方式实现对象加速，大幅降低了算法的扫描空间；

2) 算法 FARAM 在每次迭代过程中通过构造属性组的方式, 一次添加多个符合要求的属性至候选约简属性集中, 有效减少了算法的迭代次数;

3) 算法 FARAM 通过粒度比较的方式, 忽略了部分粒度较粗、分类能力较弱的属性, 有效降低了迭代过程中待评估的属性规模。

## 5. 实验分析

本节将算法 FARAM、FGARM 和算法 FSPA-PR [10]、FARA [13] 分别进行了对比。其中, 算法 FSPA-PR 和 FARA 均为针对决策表中的全部决策类进行属性约简。实验对比分为两部分: 第一部分为算法约简长度的对比; 第二部分为算法约简效率的对比, 从对象、迭代次数和约简时间三个方面进行了对比分析。

实验所采用的 6 组数据集信息如表 4 所示。实验环境描述如下: Windows 10 (64 bit) 操作系统, Inter Core i7-10750H (2.60 GHz) 处理器, 16 GB 内存, 所有算法均采用 PyCharm 2020 软件和 Python 语言进行测试。由于算法 FARAM、FGARM 均适用于多个特定决策类, 实验过程中在各数据集中选取某些特定决策类进行测试, 表中各数据集选定的多特定类决策值集合分别为: {1}、{1,2}、{1,2}、{1}、{0,1,2} 和 {1,2,3}。

Table 4. Data sets

表 4. 数据集

序号	数据集	样本数	属性数	类别数
1	Vehicle	846	18	4
2	Landsat	6435	36	7
3	Thyroid	7200	21	3
4	Coil2000	9822	85	2
5	Penbased	10992	16	10
6	Letter	20000	16	26

### 5.1. 约简长度对比

表 5 为四种算法的约简长度对比。从表中可以看出, 相比于算法 FSPA-PR、FARA, 算法 FARAM 和 FGARM 在 6 组数据集上均可获得相对较短的约简结果。相比于决策系统的全部分类, 多特定类的正域规模较小, 因此约简结果也相对较短。同时, 由于算法 FARAM 在迭代过程中一次添加多个属性且忽略了部分粒度较粗的属性, 属性的添加并不严格遵守属性重要度的大小排列顺序, 因此其约简长度要略高于算法 FGARM。比如在平均约简长度中, 算法 FARAM 的约简长度为 12.3, 略高于算法 FGARM 的约简长度 11.5。

Table 5. The length of reduction comparison

表 5. 约简长度对比

数据集	FSPA-PR	FARA	FGARM	FARAM
Vehicle	8	10	7	7
Landsat	12	13	7	11
Thyroid	13	13	12	12
Coil2000	28	29	26	27
Penbased	9	9	8	8
Letter	11	12	9	9
Average	13.5	14.3	11.5	12.3

## 5.2. 约简效率对比

本节从迭代次数、参与迭代的对象规模和约简时间三个方面进行了对比分析。

表 6 为四种算法的迭代次数对比。从表中可以看出，基于组合加速机制的算法 FARA 和 FARAM 的迭代次数相对较低。例如在数据集 Landsat 中，算法 FARA 和 FARAM 的迭代次数均为 1，而算法 FSPA-PR 和算法 FGARM 的迭代次数分别为 12 和 8；在数据集 Thyroid 中，算法 FSPA-PR 的迭代次数为 0，这是由于该算法以核属性作为约简起点，且核属性为约简结果，因此不需要参与迭代。同时，由于算法 FARA 针对决策系统中的全部分类进行约简，容易找到符合条件的属性构造属性组，因此算法 FARA 的迭代次数的迭代次数要略低于算法 FARAM。比如在平均迭代次数上，算法 FARA 的平均迭代次数为 1.6，而算法 FARAM 的平均迭代次数为 5.0。

Table 6. Iterations comparison

表 6. 迭代次数对比

数据集	FSPA-PR	FARA	FGARM	FARAM
Vehicle	7	1	7	2
Landsat	12	1	8	1
Thyroid	0	3	12	7
Coil2000	18	1	27	14
Penbased	9	3	8	4
Letter	8	1	9	2
Average	9.0	1.6	11.8	5.0

表 7 为算法 FARAM 在前四次迭代过程中所需处理的对象数目，其中“—”表示算法迭代结束，不需要处理任何对象。由于算法 FGARM 不具有任何加速机制，每次迭代都需要对论域中的全部对象进行处理，因此在表中仅表示了算法 FARAM 迭代过程中所需处理的对象数目。从表中可以看出，随着算法迭代次数的增加，算法 FARAM 会逐步地删除正域和负域中的对象，因此所需处理的对象规模也会逐步降低。若相邻两次迭代所需处理的对象数目相同，则表明该轮迭代中构造的属性组仅包含一个属性且该属性对于构造正域没有任何贡献，如数据集 Penbased 的第一轮和第二轮迭代，数据集 Coil2000 的第三轮和第四轮迭代，对象数目均没有发生改变，说明该轮迭代中选择的单个属性对于粒度的划分并无贡献。

Table 7. The number of handled objects in iterations

表 7. 迭代中所需处理的对象数目

数据集	1	2	3	4
Vehicle	846	486	—	—
Landsat	6435	—	—	—
Thyroid	7200	7200	760	49
Coil2000	9822	732	393	393
Penbased	10992	10992	7079	4001
Letter	20000	4473	—	—

图 3 为四种算法的约简时间对比。其中，横坐标为论域规模，纵坐标为算法的运行时间，单位为秒 (s)。从图中可以看出，当数据规模较小时，四种算法的运行时间都很低，但随着数据集规模的不断扩大，



四种算法的运行时间都随之增加。但是算法运行时间与论域规模并不具有严格的正相关性。比如在数据集 Vehicle 中,当论域规模由 6 增至 7 时,算法 FGARM 的时间反而有所降低。这是由于该算法为贪心算法,所选择的均为局部最佳属性,该属性仍有部分可能为冗余属性,因此会出现规模增加但运行时间反而减少的情况。对比四种算法发现,算法 FARAM 的运行时间始终保持最低,且增加幅度较小,稳定性较强。这是由于算法 FARAM 从对象、属性和粒度三个方面设计了加速机制,从多维度保证了算法的高效性,因此可较大幅度地降低约简时间。

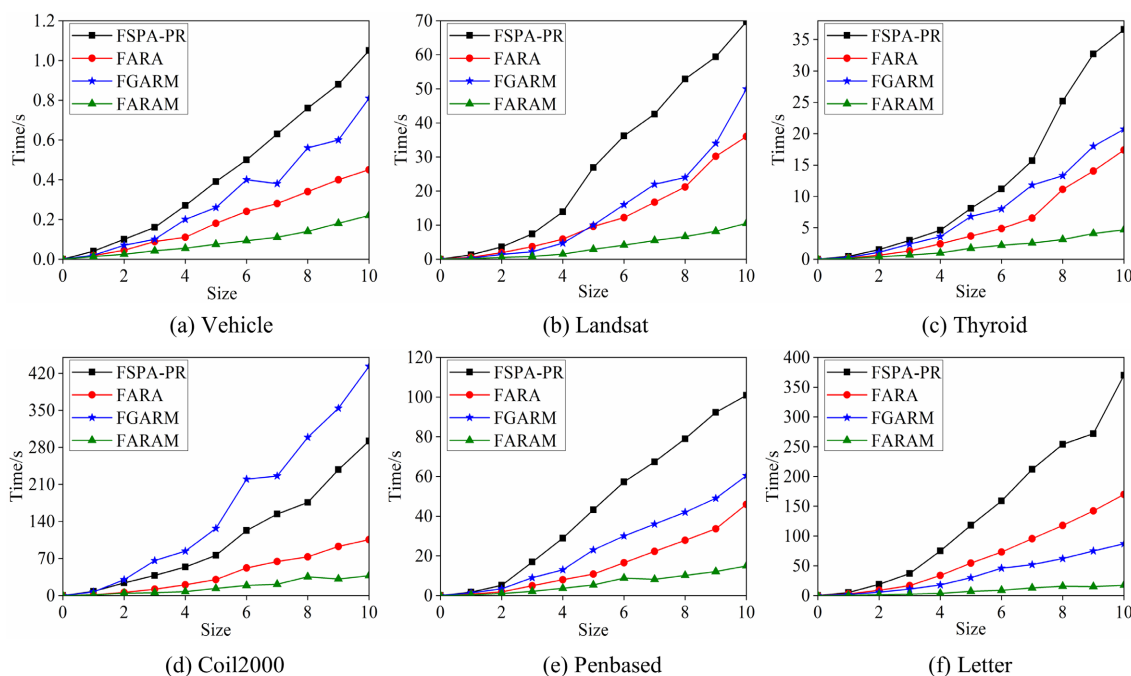


Figure 3. Reduction time comparison  
图 3. 约简时间对比

## 6. 结论

在多特定类正域约简中,启发式属性约简算法虽然能够快速地从决策系统的一个约简结果,但因其迭代过程中需要处理全部的对象和属性,而且每次迭代仅选择一个属性,所以当数据规模较大时约简效率仍然较低。本文在启发式算法的基础上,设计了基于对象、属性和粒度视角的多特定类加速方法,提出了基于组合加速机制的多特定类快速正域约简算法,通过删除正域和负域、构造属性组和粒度比较的方法,降低了参与迭代的对象规模、待评估的属性规模以及算法的迭代次数,从而提高了多特定类的约简效率。同时,由于本文在构造属性组和粒度比较的过程中,弱化了属性重要度的利用,因此约简结果可能存在冗余属性较多的情况,进一步地提高去冗余的效率是未来的研究工作之一。

## 基金项目

本文受烟台市科技计划项目(编号: 2022XDRH016)的资助。

## 参考文献

- [1] Pawlak, Z. (1982) Rough Sets. *International Journal of Computer & Information Sciences*, **11**, 341-356.  
<https://doi.org/10.1007/BF01001956>

- [2] Shu, W.H., Qian, W.B. and Xie, Y.H. (2020) Incremental Feature Selection for Dynamic Hybrid Data Using Neighborhood Rough Set. *Knowledge-Based Systems*, **194**, Article ID: 105516. <https://doi.org/10.1016/j.knosys.2020.105516>
- [3] 王国胤, 于洪. 多粒度认知计算——一种大数据智能计算的新模型[J]. 数据与计算发展前沿, 2020, 1(2): 75-85. <https://doi.org/10.11871/jfdc.issn.2096-742X.2019.02.007>
- [4] 王国胤, 代劲, 李昊. 基于多粒度认知计算的生产安全管理与决策[J]. 中国科学基金, 2021, 35(5): 752-758. <https://doi.org/10.16262/j.cnki.1000-8217.2021.05.012>
- [5] Banerjee, A. and Maji, P. (2019) Segmentation of Bias Field Induced Brain MR Images Using Rough Sets and Stomped-t Distribution. *Information Sciences*, **504**, 520-545. <https://doi.org/10.1016/j.ins.2019.07.027>
- [6] 陈超凡, 张红云, 蔡克参, 苗夺谦. 基于三支决策的二阶段图像分类方法[J]. 模式识别与人工智能, 2021, 34(8): 768-776. <https://doi.org/10.16451/j.cnki.issn1003-6059.202108010>
- [7] 钱文彬, 彭莉莎, 王映龙, 段德林. 不完备混合决策系统的三支决策模型与规则获取方法[J]. 计算机应用研究, 2020, 37(5): 1421-1427.
- [8] Yao, Y.Y. and Zhang, X.Y. (2017) Class-Specific Attribute Reducts in Rough Set Theory. *Information Sciences*, **418-419**, 601-618. <https://doi.org/10.1016/j.ins.2017.08.038>
- [9] Li, B.Z., Wei, Z.H., Miao, D.Q., Zhnag, N. and Shen, W. (2020) Improved General Attribute Reduction Algorithms. *Information Sciences*, **536**, 298-316. <https://doi.org/10.1016/j.ins.2020.05.043>
- [10] Qian, Y.H., Liang, J.Y., Pedrycz, W. and Dang, C.Y. (2010) Positive Approximation: An Accelerator for Attribute Reduction in Rough Set Theory. *Artificial Intelligence*, **174**, 597-618. <https://doi.org/10.1016/j.artint.2010.04.018>
- [11] 陈曼如, 张楠, 童向荣, 东野升龙, 杨文静. 基于多尺度属性粒策略的快速正域约简算法[J]. 计算机应用, 2019, 39(12): 3426-3433.
- [12] 赵立威, 张楠, 张中喜. 基于特征粒的序决策系统快速约简研究[J]. 山西大学学报, 2020, 43(4): 897-905.
- [13] Chen, Y., Yang, X.B., Li, J.H., Wang, P.X. and Qian, Y.H. (2022) Fusing Attribute Reduction Accelerators. *Information Sciences*, **587**, 354-370. <https://doi.org/10.1016/j.ins.2021.12.047>
- [14] Liu, G.L., Hua, Z. and Zou, J.Y. (2018) Local Attribute Reductions for Decision Tables. *Information Sciences*, **422**, 204-217. <https://doi.org/10.1016/j.ins.2017.09.007>
- [15] Zhang, X.Y., Yao, H., Lv, Z.Y. and Miao, D.Q. (2021) Class-Specific Information Measures and Attribute Reducts for Hierarchy and Systematicness. *Information Sciences*, **563**, 196-225. <https://doi.org/10.1016/j.ins.2021.01.080>
- [16] Wang, Y.B., Chen, X.J. and Dong, K. (2019) Attribute Reduction via Local Conditional Entropy. *International Journal of Machine Learning and Cybernetics*, **10**, 3619-3634. <https://doi.org/10.1007/s13042-019-00948-z>